DOCUMENT RESUME

ED 163 973 IR 006 646

AUTHOR	Blaine, Lee; And Others
TITLE	Technical Problems in Implementing University-Level Computer-Assisted Instruction in Mathematics and Science: First Annual Report. Technical Report No. 293.
INSTITUTION	Stanford Univ., Calif. Inst. for Mathematical Studies in Social Science.
SPONS AGENCY	National Science Foundation, Washington, D.C.
PUB DATE	21 Apr 78
GRANT	SED-77-09698
NOTE	29p.; For related documents, see IR 006 644-647
EDRS PRICE	MF-$0.83 HC-$2.06 Plus Postage.
DESCRIPTORS	*Computer Assisted Instruction; Computer Programs; Computer Science; Data Processing; *Educational Research; Mathematics Instruction; On Line Systems; Programing Languages; *Technical Mathematics; *Universities
IDENTIFIERS	*VOCAL

ABSTRACT

Difficulties in implementing the EXCHECK/VOCAL System, a general program for mathematics instruction written in the VOCAL language, are presented in terms of informal mathematics procedures, audio and prosodic features, and proposed research. References are appended. (CMV)

TECHNICAL PROBLEMS IN IMPLEMENTING UNIVERSITY-LEVEL COMPUTER-ASSISTED INSTRUCTION IN MATHEMATICS AND SCIENCE: FIRST ANNUAL REPORT

by

Lee Blaine, Arvin Levine, Robert Laddaga, and Patrick Suppes

NSF Grant No. SED77 - 09698

TECHNICAL REPORT NO. 293

April 21, 1978

PSYCHOLOGY AND EDUCATION SERIES

INSTITUTE FOR MATHEMATICAL STUDIES IN THE SOCIAL SCIENCES

STANFORD UNIVERSITY

STANFORD, CALIFORNIA 94305

Acknowledgments

The authors wish to thank the following people for their contribution to this report: Lawrence Zaven Markosian, James McDonald, and William Sanders.

For their contributions to the research conducted at the Institute and described in this report we also thank: Barbara Anderson, George Black, Edward Bolton, Thomas Dienstbier, Pentti Kanerva, Ingrid Lindstrom, Sten Lindstrom, Teri Pettit, Ronald Roberts and Marguerite Shaw.

The work described in this report was supported by the National Science Foundation under NSF Grant No. SED77-09698.

Table of Contents

1 Informal Mathematical Procedures

The work on informal mathematical procedures is being carried out as outlined on page 82 of the original project proposal. We quote from that page:

> "During the first year, the EXCHECK proof checker will be redesigned not only to function as an interactive proof checker, but also to function as an interactive theorem prover capable of accepting and executing complex proof strategies. ...Also in the first year, the mechanisms to implicitly handle sorts will be expanded to handle types. The resolution theorem prover will be modified to make it recognize more proofs that are obvious to the student. ...The new introductory lessons will be begun in the first year...The work on the help system will be mostly done in the first year... Generally, during the three years covered by the proposal, the search for ways of informalizing the EXCHECK system will be continued."

The work on theorem provers is described in Section 1.3, and the work on types is described in Section 1.4.2. Section 1.5.1 describes the work carried out on the HELP system and the introductory lesson. General system level work, needed to implement the various changes to EXCHECK described in this report, are outlined in Section 1.1. Section 1.2 and Section 1.4 discuss work done to informalize the EXCHECK system.

1.1 General System Work

The compiler for the VOCAL author language, the theory processor, the curriculum driver, the proof checker, and various auxiliary programs were restructured in order to create more uniform modules out of which the programs could be constructed. This has resulted in a considerable savings in runtime space usage as well as making it easier to generate

and test new versions of the programs. New methods for specifying the parameters of the various theories were implemented making it easier and faster for curriculum authors to specify theories and also resulting in a faster runtime system.

The basic programing language (SAILSP--for SAIL with a version of LISP embedded in it) used in the EXCHECK system was modified to reduce explicit dependence on the LEAP features of SAIL thus facilitating the restructuring mentioned above. The LISP garbage collector in SAILSP was improved and methods for more efficient transfer of s-expressions from file to core and from fork to fork were implemented.

1.2 More Natural Semantics

One of the most difficult problems in writing CAI programs is developing procedures that permit the interaction to take place at a natural semantic level. A large part of earlier work on the EXCHECK system was directed specifically at this problem. The IMPLIES rule is a good example of this earlier work. It uses extensive computation and heuristics to ensure that the user is not distracted by mathematically irrelevant logical detail.

The BOOLE procedure is a good illustration of another method of placing the interaction at a natural semantic level. BOOLE uses a decision procedure to determine if a given formula is a theorem of quantifier free boolean algebra. Hence, students in set theory and other courses can use it to assert that such formulas are true.

Work has continued during this last year on facilitating interaction on a semantic level. The REPLACE rule and the ABBREVIATION procedures were completely rewritten for this purpose. Other

procedures, particularly IMPLIES and VERIFY, were altered to make them correspond more closely to natural inference. The ELFS procedures described below were implemented as part of the program of extending the nonderivation exercises. Also the VOCAL language is being considerably extended by the graphics and speech oriented semantics package described below.

1.2.1 ELFS

'ELFS' is an acronym for Explicitly Listed Finite Sets. The name comes from the fact that users must specify the basic sets by explicitly listing their elements. In the current implementation the structure dealt with is the class algebra over the set {0,...,35}. The language handled is the quantifier free language for this structure. There are procedures available to determine the denotation of any term and the truth value of any formula in this language (assuming of course that the denotations of variables has been determined).

In a typical exercise the student is asked to specify ELFS that satisfy a given condition. For example the student might be asked to give sets A and B such that the cartesian product A X B is distinct from B X A. This is equivalent to: A is not equal to B and both are nonempty. Given values for A and B the ELFS procedures can easily compute the truth or falsity of the latter formula. Also because these procedures are available in the answer analysis routines of VOCAL the author can easily formulate semantically based responses to student responses. Continuing the example, an author might say in effect: if one of the sets A, B given by the student is empty ask the student 'what is the cartesian product A X B if either A or B is empty?'.

The ELFS represent an important extension of the EXCHECK/VOCAL CAI system because they provide a semantic basis for the illustration and interactive discussion of mathematical concepts. Previously the principal constructed responses allowed students were derivations. This limitation placed a premium on skill in constructing derivations, which is only one component of the mathematical maturity that the CAI courses strive to develop.

1.2.2 Extended Graphics and Speech Oriented Semantics

We are currently implementing a graphics and speech oriented semantics package for the EXCHECK/VOCAL CAI system to give authors better facilities for presenting material to students informally, precisely, and at a natural semantic level using a variety of modes. In particular it is now easier for authors to illustrate mathematical concepts and to specify semantically appropriate responses to student responses. Fundamentally this is accomplished by providing facilities for the author to specify how formulas and terms of languages such as the languages for set theory and logic can be "interpreted" visually and aurally.

For example, using the semantics package, a graphic representation of a fact about sets can be displayed for the student. At the same time, audio comments can be made using the computer synthesized speech system. This combination is especially useful for introducing new concepts and for testing the student's understanding of previously introduced concepts.

The semantics package is designed to be versatile and extensible. Not only can the semantics of languages be specified but also the basic

display and audio commands available for constructing semantics include most of the VOCAL commands already available to the curriculum author. Among these basic commands are those which display, erase, brighten and unbrighten areas on the video-display screen, and those which generate audio messages. Other VOCAL commands available for constructing semantics include some which require student interaction. So, for example, it is now possible to display the visual semantic values of the subterms of a complex term separately, each followed by a pause requiring a student response before continuing. As a further example, using the semantics package authors can easily create exercises that require the student to input a term in an appropriate language. The student's response then can be displayed and compared graphically with correct solutions.

1.3 Theorem Provers

Implementation of the new interactive natural deduction theorem prover has begun and the first version should be finished during the first quarter of next year. Most of work so far has gone into designing program control and data structures adequate for an interactive natural deduction theorem prover and yet efficient enough for real time use.

As an example, a significant operation in the new prover is the natural deduction analog of unification: comparing two formulas to determine which (possibly empty) set of instantiations would make the formulas equivalent. The procedure that does this is called INSTANCE. In resolution provers the formulas passed to unification are in a rather simple canonical form. In the new prover INSTANCE is applied to formulas in standard form. Hence there is a question of time

efficiency. We have been able to design data structures that permit INSTANCE to run efficiently on formulas in standard form. In particular, INSTANCE without identity runs in a time proportional to the lisp EQUAL function without requiring that biconditionals, uniqueness quantifiers, or other complex structures be converted to a more primitive form. The complexities and increased running time introduced by adding identity to INSTANCE seem no worse than for analogous procedures such as resolution and in fact might be significantly more efficient although proofs establishing this have yet to be done. The general data structures are also designed to be efficient for identity. In both cases the design grew out of our work on the TEQ algorithm, work done with resolution theorem provers, and consideration of various back-tracking and parallel processing schemes.

Substantial effort has also been put into designing top level data structures and program control capable of efficiently interacting with users and accepting complex proof strategies from them. A major problem here is structuring program control so that interaction can occur for global or strategic decisions about how to do the proof while leaving the procedures that handle routine work to function without interruption. The current version employs a multilayered approach to handle this problem. The user interacts with the STRATEGIST which in turn controls the routine work via a subordinate procedure: the TACTICIAN. Another beneficial aspect of this design is that, very roughly, decisions made by the TACTICIAN are local and have linear effect while decisions made by the STRATEGIST are global and can have more than linear effect. The multilayered design also provides a natural basis for the acceptance and execution of complex strategies:

they are passed directly to the STRATEGIST which in turn uses them to direct the TACTICIAN in its work.

Most of this year's work on theorem provers has been devoted to the strategic theorem prover, although the current resolution prover has been maintained and data collected for use in the development of any new provers. Near the beginning of this year, work on a substantial revision of the resolution prover was done in parallel with early work on the strategic prover. Most of the results of that work will be incorporated in routines in the strategic prover.

1.4 More Informal Mathematical Procedures

1.4.1 Informal Languages and Proof Summarization

The output grammar is being extended to include a more informal mode. For example, the extended procedures will output `f maps A into B' instead of `f: A -> B' and `every element of B is an element of C' instead of `(A x)(if x in B then x in C)'.

Procedures are being developed that will permit effective and natural summarizations of derivations. The procedures first organize the parts of the proof on the basis of logical dependencies and then summarize the resulting parts on the basis of the major steps. When the new informal mode of the output grammar is completed it will be coupled with the proof summarization procedures to produce informal summaries of proofs.

1.4.2 Sorts and Types

Work has begun on the extension of the sort machinery to handle the "types" of functions; i.e., to extend the implicit sorting machinery to

compute not only the sort of object but also to compute descriptions for function that code how to compute the sorts of their values given the sorts of their arguments. This extension is quite difficult to accomplish efficiently as the introduction of a new type for a function could require updating the sorts and types of a great many objects.

1.4.3 Simplification and Computation

Work on providing students with more access to the simplification procedures available through the REDUCE fork of the EXCHECK system is still in the experimental stage. Currently, students can use the ALGEBRA command to manipulate identities.

An Example of the Use of the Current Implementation of the ALGEBRA Rule:

(8) P(A) + P(B) = P(A U B)

*8alg$EBRA
Schematic form:
r+t = x1

New form *t = x1 - r$

Multiplicative factor used to get new form *$
8 ALGEBRA
(9) P(B) = P(AUB) - P(A)

note: underlining indicates student input, $ denotes 'ENTER' key.

It would be of great value in the course in the foundations of probability to extend the algebra rule to handle cases involving inequalities. Thus, it would facilitate inferences such as the following:

1. $q-t \leq r-t$ implies $q \leq r$
2. $q \leq 0$ & $r < 0$ implies $q/r \leq 0$
3. $q \leq r$ & $q > 0$ implies $r > 0$

Complex calculations involving differentiation and integration would also be of great use in the course, and could be implemented in the REDUCE fork (see Hearn for a description of the REDUCE language). The student should also be able to do very simple calculations in preparation for responding to short answer questions. Rather than generate a call to the REDUCE fork we will incorporate a calculator into the upper fork of the probability course. One example of the kind of calculations involved is the following.

Students are asked to compare (first intuitively) the probability of getting at least one 6 in four throws of a die, to the probability of throwing at least one double-six in twenty four throws of a pair of dice. People will often assume that the probabilities will be the same, especially if is pointed out that 4/6 equals 24/36. Afterwards the student can calculate for himself precisely what values the probabilities in question have. They are:

$$
\begin{aligned}
P(6 \text{ in four throws}) &= 1 - P(\text{no } 6 \text{ in four throws}) \\
&= 1 - P(\text{no } 6 \text{ in one throw})^4 \\
&= 1 - (5/6)^4 \\
&= 0.518
\end{aligned}
$$

$$
\begin{aligned}
P(6{,}6 \text{ in } 24 \text{ throws}) &= 1 - P(\text{no } 6{,}6 \text{ in } 24 \text{ throws}) \\
&= 1 - P(\text{no } 6{,}6 \text{ in one throw})^{24} \\
&= 1 - P(35/36)^{24} \\
&= 0.491
\end{aligned}
$$

1.4.4 Interplay Between Theoretic and Metatheoretic Methods

In the proof theory course results are established in a metatheory (TEM) for Zermelo-Frankel (ZF) set theory. In particular, students have

to establish in the metatheory that under given conditions certain results are provable in the object theory. The standard mathematical method for doing this is to simply derive the result directly in the object theory and then use this fact in the metatheory.

Procedures have been added to allow students to make such inferences in the proof theory course. Two inference procedures are involved: ZFSTART for starting a derivation in ZF from the metatheory, and ZFFINISH for finishing the derivation in ZF and returning to the metatheory. After starting a ZF derivation from the metatheory prior results from the metatheory or the metatheoretic part of the derivation may be referenced from the ZF part of the derivation. In the example below, two lines in the metatheoretic part of the derivation are referenced from the ZF part of the derivation. There is a restriction on the form of metatheoretic results that may be referenced from inside ZF: they must be atomic formulas of the form ZF |- F or conjunctions of such formulas.

```
Derive:
IF  ZF |- x=y AND  ZF |- y=z THEN  ZF |- x=z


WP      (1)  *ZF |- x=y
WP      (2)  *ZF |- y=z
*zfs$TART
                   *****************  ZF  *****************

   #1,2teq$  (3)  *x=z$
Will you wish to specify? (No) *$
Using *g$O
   #3zff$INISH
                   ****************************************

3,1,2 ZFFINISH
        (4)   ZF |- x=z
```

1.5 User Aids

1.5.1 HELP System and Introductory Lesson

The introductory lesson for the EXCHECK courses was completed this summer. Student documentation was updated and improved, to reflect the changes in the EXCHECK system. Previously, the HELP system was loaded into the program only upon being called by the student. This resulted in long waits for the HELP files to be read in. The system was revised to permit loading with the EXCHECK program, thus reducing the wait during a students first call to HELP. The strongest barrier to extending the HELP system to its full potential currently is the need to produce remedial, enrichment, and tutorial curriculum modules to be accessed by the system.

1.5.2 Shorthand for entering Terms and Formulas

A new notation system has been added for referring to parts of already accepted terms and formulas. This system provides students a shorthand method for entering terms and formulas.

For example, assume that a student is doing a derivation which contains the following step:

```
(4) A is a subset of {E: E is in pow(C~D)}
          iff
     A is a subset of f(G)
```

The student could enter the formula

```
not ( A is a subset of {E: E is in pow(C~D)}
          and
       A is a subset of f(C) )
```

by typing: not (fm:4:1 & fm:4:2). The expression 'fm:4:1' designates the <u>formula</u> that is the first subexpression of the formula on line 4.

Similarly, `tm:12:2:1' would designate the term that is the first subexpression of the of the second subexpression of the formula on line 12. Designators begining with `fm' are parsed by the grammar as formulas and hence can be used in any expression at any point at which a formula is appropriate. Similarly, designators begining with `tm' can always be used instead of the term designated.

2 Audio and Prosodic Features

The work on Audio and Prosodic Features is proceeding largely as outlined on page 83 of the original project proposal. We quote from that page:

> "In the first year of this proposal, we would design and build the MINI-MISS machine, implement some of the data-compression techniques, and do some preliminary studies on the quality of the audio we are producing. We would continue improvements to prosodic contours and the syntactic parser and add prosodic contours. In particular, the syntactic parser and contour-generation producers would be reimplemented for efficient on-line generation of synthesized speech.
>
> In the second year, we would ... simulate the concatenation of affixes to root words."

The work done on data-compression techniques in connection with the design of the MINI-MISS machine is described in Section 2.1. The preliminary studies of prosodic quality are discussed in Section 2.5. We also discuss improvements to the syntactic parser (see Section 2.2) and prosodic contours (see Section 2.3).

As stated above, we expected to delay work on an algorithm for concatenation of affixes to root words until the second year of the grant and to emphasize on-line prosodic generation in the first year. As explained in the interim report, it was necessary to exchange the two. Our work this year on concatenation is described in Section 2.4, and our plans for on-line generation of prosody are discussed in Section 2.2.2.

2.1 Data Compression Techniques

The present MISS synthesizer is not suitable for duplication for use at sites remote from Stanford. The use of LSI integrated circuits and the removal of the generalized features that are necessary for speech research will allow us to build a MINI-MISS machine which would fulfill the requirements of a small cluster of remote terminals for using audio. The speech parameters would be transmitted to this MINI-MISS machine via the same data-line connection that the terminals would use, although that data line would require a higher bandwidth to handle the additional speech-parameter load. The same high speech quality and prosodic manipulations available in the full MISS system would be provided to such remote users with the MINI-MISS machine.

Integral to the design of the MINI-MISS machine, we are attempting to utilize the results of recent research in speech synthesis. In particular, new techniques for improving the quality of the synthesized speech and decreasing the storage size required to represent the speech have been extensively studied. We have been duplicating these experimental findings in the context of the MISS system. The importance of conducting these studies before committing ourselves to a design for the MINI-MISS machine can be demonstrated with respect to two specific issues, log area ratios and covariance analysis. The use of delta-encoded log area ratio reflection coefficients holds great promise in substantially reducing our storage requirements and the transmission rate for the MINI-MISS machine. The digital filters required for audio synthesis using these coefficients is, however, incompatible with those designed for the LPC parameters we are currently using. Thus it is critical to evaluate the potential usefulness of these coefficients

before designing and specifying one of the central parts of the new machine.

Similarly, the use of covariance linear predictive analysis instead of auto-correlation linear predictive analysis would improve the quality of the synthesized speech. The covariance techniques sometimes generate instabilities in the speech parameters, which, unless overcome, would make this technique unsuitable for our operational synthesis system. We have found that the literature on covariance analysis understates the importance of these instabilities, and we are experimenting to discover if they can be minimized sufficiently. Again, in this case, the design of the new machine is dependent on the results of our experimentation and duplication and evaluation of current research results.

2.2 Improvements to the Syntactic Parser

We have increased the complexity of the syntactic sub-component of the prosodic system in order to expand the range of constructions which we can accommodate. Most of these extensions are not notable as discoveries of new ways of parsing but are rather the outgrowth of selected review of the structures that were generated by the previous parser. One extension that is interesting is the recognition of different grammatical functions for words in some word classes. In addition, streamlining the syntactic sub-component will be useful in the on-line generation of prosodic contours.

2.2.1 Multiple word functions and homographs

There are many cases of homographs, words spelled the same, but pronounced differently and having different grammatical functions, in

English. Consider the case of two tenses of the word "read". In the present tense, the word is homophonous with "reed", while in the past tense it is homophonous with "red." A large class of adjectives and verbs with identical spellings differ in pronunciation depending on which grammatical function they fulfill. Many nouns and verbs also have this property. Some examples of adjectives and verbs with identical spellings are:

> present (adj: current, vb: to give)[1]
> alternate (adj: other, vb: to switch between 2 choices)
> compact (adj: mathematical concept applies, vb: to compress)

Examples of nouns and verbs are:

> affix (noun: prefix or suffix, vb: to add something to a base)
> content (noun: material contained, adj/vb: happy/to make happy)

We have improved our syntactic analysis so that it chooses correctly in most cases the correct grammatical function and consequently the correct sound for these homographs. In most cases, we have also had to create one of the two forms of the spoken word. This aspect is discussed more fully below, Section 2.4.

2.2.2 On-line generation

While we had initially expected that a greater importance would be placed on the development of an on-line generation algorithm than on word formation, the actual course of research has exchanged the two. Our preliminary plans for on-line proof summaries in audio are to precompile, in parameterized form, the messages to be generated. For example, consider a message type such as "alpha proves that beta holds when gamma is also true", where alpha, beta and gamma are place holders for potentially complex phrases. A parameterized form of synthesizer

[1]This word can also be a noun, meaning 'a gift'.

commands that will generate correctly the static parts of the message and give a good approximation for speaking the specific material filling in the place holders will be stored with the routine for selecting the message. During the next year of the grant, we will implement this routine and evaluate it, trying to estimate the cost in processing time resulting in various grades of audio quality.

2.3 Prosodic Contours

The major development in our prosodic contouring was the implementation of ratio intervals on the pattern of musical notes for the peak pitches which we assign to each word in an utterance. Fuller discussion of this idea is presented in Levine (1977), but the simple idea is that the targets for the pitches in an utterance take the relationship of musical intervals to each other. It has been noticed (Liberman, 1975) that in children's chants, the intervals between words are approximately minor thirds. We have carried this idea further and are using ratios ranging from a quarter tone to a perfect fifth and octave in generating our prosodic parameters. Using musical intervals does not mean that the words are sung. We expect that the musical intervals sound more natural than the previous assignments that we were using.

We are currently trying to experimentally determine whether there is a significant improvement in student acceptance of this form of prosodic generation over the previous (non-musical) system. The actual difference in values of parameters is quite small, and, while the new version did seem to us to be qualitatively better, we have not yet been able to adequately prove our point. The proof is especially elusive

since the complex factors affecting listener acceptance of this speech prevent us from isolating the response to the particular phenomenon we are investigating.

We also presented some results, comparing our audio system with other potential and actual systems, at the Acoustic Society of America, 94th meeting.[9] One paper showed that our word-based prosodic model corresponds well to theoretical predictions developed by Klatt (1975). A second paper compared the rate at which school aged children are able to adapt to synthesized speech.

2.4 Word formation by concatenation

In the first year of the grant we have begun to develop an algorithm that will automatically concatenate roots and affixes in order to produce words which were not recorded for our lexicon. There are two situations in CAI where such 'new' words are required. One is for meeting the needs of curriculum authors who may modify existing lessons or design new ones which may introduce new terms or use forms of words which were not in the old curriculum. We mentionned above the need for having both spoken versions of homographs available to the syntactic parsing system. Since the lexicon initially contained only one spoken version of each word we have had to form the other ourselves. The other need comes from interactive prosodic generation. In its most general form, the interactive component will use words which may not have been in the curriculum, in order to respond utilizing the student's own terms and phrases. Our approach to the production of new words has been to develop progressively automated techniques for concatenation. The more rudimentary techniques aid us in expansion of the lexicon for use by

curriculum authors in modifying lessons or producing new ones. Our current procedures are suitable for inclusion into the next, more complex, stage of concatenation.

2.4.1 Research on Concatenation

Our first step in this research was to duplicate the initial research done by Lovins and Fujimura. This work uses a special set of "demisyllables", each sounding like half a syllable, which can then be concatenated with each other to form a regular syllable or word. For example, the "bi-" taken from the "bid" might be combined with the "-it" taken from the word "sit" and concatenated to form the word "bit." We have based our general procedure on this concept with the exception of certain affixes (such as the plural and "-ed" endings) which are better treated as combining with a full root word whose last segment has been slightly shortened.

One problem that became apparent when we implemented this approach is that since there is a need for a linear smoothing of the LPC coefficients at the concatenation boundary, we have to be careful about the resulting segment amplitude. The coefficient smoothing creates a "new" segment at the boundary as part of the smoothing process, but we do not yet have a good predictor of loudness for that created segment. Thus we are subject to the introduction of clicks and pops at the boundary. We are experimenting with an alternative approach to amplitude for incorporation into the MINI-MISS machine, which should simplify this problem.

Another aspect of concatenation that needs to be considered is assuring that the resulting word will have the appropriate duration. We

are currently examining the algorithm from Klatt (1975), in this respect. We are also developing automatic procedures for identifying syllable boundaries. These boundaries are critical for the development of the set of demisyllables, and usable root words. Currently, we use a measure of spectral distance, or how constant the sound parameters are, to determine where the vowels and syllables are in each word, but this attempt will certainly need to be complicated to include more sophisticated linguistic information.

2.4.2 Additions to the Lexicon

While developing the automated procedures for word formation, we are already using the sub-procedures at hand to form the words which various courses required. In particular, in the first year, almost 200 words were formed by hand, using these sub-procedures. In general the method has been to start with a sequence of words in the vocabulary which will each contribute a part of the desired word. An automatic sub-procedure is used to create the appropriate demi-syllable boundaries for each word in the sequence. The boundaries then may be adjusted by hand to improve the concatenation. Finally the actual concatenation, with pitch, duration and loudness adjustments (but without segmental smoothing at the boundaries) are made. The newly formed word is stored in the general vocabulary, but is marked as specially formed. Once in the vocabulary, the word is used just like any other LPC analyzed word, for sentence generation.

2.5 Study of Prosodic Quality

During the course of this year we have been gathering data on

comparison of student preference for audio with stored digitized phrases as against audio with automatically generated prosodic contours. The experiments involve both direct comparison, and indirect comparison via a third nonaudio mode. Preliminary results seem to indicate that student preference for audio as against non-audio mode is no less when when the audio mode is the synthetic prosody, than when the audio is canned phrases.

Further data is being gathered during the current quarter of the introductory logic course. More detailed analysis will be carried out on this years data during the final quarter of this year of the grant and in the first quarter of the next year.

3 Schedule of Proposed Research

3.1 Informal Mathematical Procedures

The schedule for completing research originally proposed for the second year of the grant is essentially the same as that given on page 82 of the original technical proposal.

The primary goal for the second year is to finish implementation of the first version of the interactive natural deduction theorem prover and to refine and extend it on the basis of data from use in the courses. The basic language for stating the proof strategies to be accepted by the prover will be implemented and informal languages for proof strategies will be investigated (see pages 41-50 of the original proposal for details). The basic interactive mechanisms for describing partial proofs to students and receiving further instructions will be implemented (see page 50 of the original proposal). Refinements of these methods will be studied and evaluated on the basis of data collected. Procedures for giving dynamic guidance to the student based on how the natural deduction prover would continue the students current work will be investigated and if possible implemented at the end of the second year.

The computation procedures available to curriculum authors will be extended in order to broaden and deepen the range of possible exercises. Particular attention will be given to assuring that student interaction with the system during these exercises is at a natural semantic level. In particular, mechanisms will be added to evaluate expressions in particular structures or theories (where feasible of course). For example, truth or validity in class algebras over finite domains, small

finite structures with a binary relation, etc. For further details see pages 39-40,55 of the original technical proposal.

The general work on informalization of the inference machinery and placing the user/system interaction at a natural semantic level will continue. In particular, work on informal output languages and informal proof summarizations will be continued, as described on pages 39-40,51-54 of the original proposal.

Also general system work aimed at improving space or time efficiency will be continued as will such work aimed at facilitating use of the system by curriculum authors (see pages 39-40,53,56-57 of the original proposal).

3.2 Audio and Prosodic Features

The schedule for completing research originally proposed for the second year of the grant is substantially the same as that given on pages 83 of the original technical proposal, with one exception. Work on concatenation of affixes and roots, originally proposed for the second year, was moved to the first year in order to meet the need for more vocabulary in the dictionary of stored sounds. The reasons for this change are discussed in the semi-annual interim report. Because of this change, we have rescheduled the efforts to implement efficient on-line generation of synthesized speech, as described in pages 63-65 of the original proposal.

In the next year of this grant we will also test the remote MINI-MISS machine in the field, and continue to implement new LPC techniques (see pages 58-59 of the original proposal). We will continue to study the relation of the quality of the synthesized speech as described on

pages 65-66 of the original proposal. The semantic analysis and story-view analysis (see pages 61-62 of the original proposal) will be added to the total prosodic analysis. Work will continue on concatenation and improving the algorithm for digitizing individual words (see page 63 of the original proposal).

References

1. Hearn, A. C. REDUCE User's Manual. Salt Lake City, Utah: University of Utah, Department of Computer Science, 1973.

2. Hinckley, M., Laddaga, R., Prebus, J., & Smith, R.L. VOCAL: Voice Oriented Curriculum Author Language. Stanford, Calif.: Stanford University, Institute for Mathematical Studies in the Social Sciences, 1976.

3. Klatt, D. H. Linguistic uses of segmental duration in English: Acoustic and perceptual evidence. Journal of the Acoustical Society of America, 1976, 59(5), 1208-1221.

4. Levine, A. Automatic generation of intonation in English. Doctoral dissertation, Stanford University, 1977.

5. Liberman, M. A. The intonational system of English. Unpublished doctoral dissertation, Massachusetts Institute of Technology, 1975.

6. Lovins, J. B., & Fujimura, O. Synthesis of English monosyllables by demisyllable concatenation. Journal of the Acoustical Society of America, 1976, 60(Suppl.No.1), S75. (Abstract)

7. Sanders, W. R., Benbassat, G. V., & Smith, R. L. Speech synthesis for computer assisted instruction: The MISS system and its applications. SIGCSE Bulletin, 1976, 8(1), 200-211.

8. Smith, R., & Blaine L. A generalized system for university mathematics instruction. In R. Colman & P. Lorton (Eds.), Computer science and education, Proceedings of the ACM SIGCSE/SIGCUE Joint Symposium, 1976.

9. Suppes, P. Technical Problems in Implementing University-level Computer-assisted Instruction in Mathematics and Science. A research proposal submitted to the National Science Foundation, 1977.